An Ontology for Semantic Web Services

Qizhi Qiu and Qianxing Xiong

Wuhan University of Technology, Wuhan, P.R. China
{qqz,xqx}@whut.edu.cn

Abstract. An ontology for Semantic Web Services is proposed in this paper, whose intention is to enrich Web Services description. Distinguish from the existing ontologies, the proposed ontology is based on both functionalities and performances, and it is organized as a layered construction. The discovery related with the proposed ontology is also discussed. Based on the Service Oriented Architecture, the proposed ontology is helpful for requesters to find their suitable services according to their own preference. Besides, as an example, an ontology for the learning resource is organized in the paper.

1 Introduction

As a novel Internet component, the significance of Web Services means enhanced productivity and quality by reuse and composition of Web Services. Developers use the services as fundamental elements for applications. As an emergent paradigm, Services Oriented Computing (SOC) is based on the service provider and service requester, where the provider builds a set of invocable applications that provide certain functionalities, while the requester invokes the services or composes suitable application.

With the development of network, there are more and more Web Services that are on different software and vendor platforms, and how to discover and compose them is one of the most challenges. All kinds of network environments require effective discovery and composition to satisfy the different kinds of requirements of requesters.

1.1 Existing Architecture—Service Oriented Architecture (SOA)

There are three main roles in SOA, which are services provider, services registry and services requester. In order to implement the basic operations (publication, discovery and binding), SOA should describe the services explicitly. The service descriptions are organized as a Service Directory. For example, UDDI (University Description, Discovery and Integration) Registry is a well-known tool.

However, UDDI organizes the services according to the defined categories (including the built-in NAICS, UN/SPSC and the user defined ones) without any semantic analysis, such as content or context. UDDI's search is based on syntax and relies on XML, which also enables syntactic. Syntax-based matching limits the reuse of Web services provided by the different kinds of providers. As we know, content and context play the important roles during the dynamic discovery and composition process, and semantic-based matching allows the requesters query through the content and context.

R. Perrott et al. (Eds.): HPCC 2007, LNCS 4782, pp. 776–784, 2007.

Furthermore, UDDI doesn't provide the quality description so that it can't enable the requesters to select the suitable ones. One of the paper's intentions is to enhance the semantic capabilities of UDDI by using tModel and so on.

1.2 Semantic Web Services

Fillman states that the Semantic Web intents to generate Web pages or services with formal declarative descriptions that programs can use to find the appropriate service and use it correctly [1]. To enrich Web Service description, the set of methods and tools in Semantic Web are employed. Ontology is an important part in the semantic web architecture proposed by Tim Burners-Lee in 2000 [2]. OWL-S is a well-known technology built inside the Web Services, and it is an ontology for web services that supplies Web Service providers with a core set of markup language constructs for describing the properties and capabilities of their Web Services in unambiguous, computer-interpretable form [3].

However, OWL-S also depends on WSDL (Web Services Description Language), an XML format for describing network services [4]. As one of main areas of Web service protocol stack, WSDL is used for describing the public interface to a specific web service. When OWL-S is introduced to describe services, the transformation from OWL-S to WSDL occurs. And there are semantic losses in the mapping stage. That means WSDL doesn't keep the rich semantics of OWL-S.

Although there are different ways to enhance the discovery of Web Services, there are several limitations in its description: (i) lack of semantics, such as pure WSDL, (ii) lack of non-functional attributes. All the above also bring forward new challenges to traditional SOC. The intention of this paper is to propose an ontology for Semantic Web Services in SOA so that UDDI can discover the more suitable services according to the preference of the requesters.

2 Ontology for Semantic Web Services

Ontology is a modeling tool for conceptual model, which describes the information system in terms of semantics and knowledge. The following is famous Guarino's definition of ontology: [5]

> *An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world.*

The definition shows that the intention of ontology is to capture, describe knowledge in explicit way, where the domain knowledge is represented as concepts and relationships. As a result, knowledge sharing is attained.

2.1 Ontology for General Purpose

Definition 1. Ontology O is defined as:

$$O = <c, p, a>$$

Where:

- *c* is a set of name concepts,
- *p* is a set of property concepts,
- *a* is a set of axioms.

In Definition 1, the axioms are the constraints and rules that are defined on the concepts and properties. There are four types of relationships between the concepts, which are part-of, kind-of, instance-of and attribute-of.

The triple in Definition 1 is corresponding to OWL (not OWL-S). The set *c* is equal to *Class* in OWL, whose elements are URI or expressions. The set *p* is equal to *Attributes* so that all the data types in XML Schema can be used in OWL.

In a word, *Ontology* is a static conceptual model for domain knowledge, which uses terminologies and their relationships agreed upon by wide communities to describe the domain knowledge and its structure.

2.2 Ontology for Web Services

Definition 2 (Web Services description model). A Web Service is defined as:

$$Service\ (d,\ f,\ p)$$

Where:

- *Service* is a Web Service that supports semantics;
- *d* is the basic information about *Service*, including name, ID, provider ID, taxonomy, version, etc.
- *f* is the functional information about *Service*, including related properties, the set of input/ output data and data-type, etc.
- *p* is the performance information about *Service*, including concerned non-functional properties.

At present, almost the web services descriptive languages are compliance with the model spontaneously, but they are all one-sided. For instance, WSDL addresses to describe the online communications in a structured way by defining an XML document for network services. It cannot deal with the semantic heterogeneity and not support the description for constraints that are important to composition.

As semantic markup for Web Services, OWL-S defines an upper ontology for services. Each instance of service will present a *ServiceProfile* description, be described by a *ServiceModel* description, and support a *ServiceGrounding* description. These descriptions answer respectively “what does the service provide for prospective clients?”, “how is it used?”, “how does one interact with it?”. Besides the semantic loss that is mentioned above, OWL-S is regardless of quality of service [6]. For these matters, when semantic web service is proposed, there are several problems must be solved: (i) ontology-based representation, (ii) quality model, (iii) support of discovery and composition. The remainder of this paper will discuss the related issue.

2.3 A Layered Construction

According to the ontology methodology, certain service ontology can be organized into a layered construction (Shown in Fig. 1):

- Top level: there is the general knowledge or common sense, independent of any domain. In a word, they are shared by wide communities. For example, NAICS (The North American Industry Classification System) is a new industry classification.
- Domain knowledge level: there are domain ontology and upper ontology. The upper ontology is defined by *Definition* 2, which is a model to describe the attributes and operations of web service. Domain ontology aims to serve for the providers who are in certain domain. For example, a travel service. Both domain ontology and upper ontology is not invocable, just like an abstract class. What they provide is a framework for a concrete, invocable service. This level can also be called as a semantic level.
- Concrete level: there are invocable services that are described by OWL and organized according to the framework defined in the upper level. Besides, there are bindings to the specific implementations.

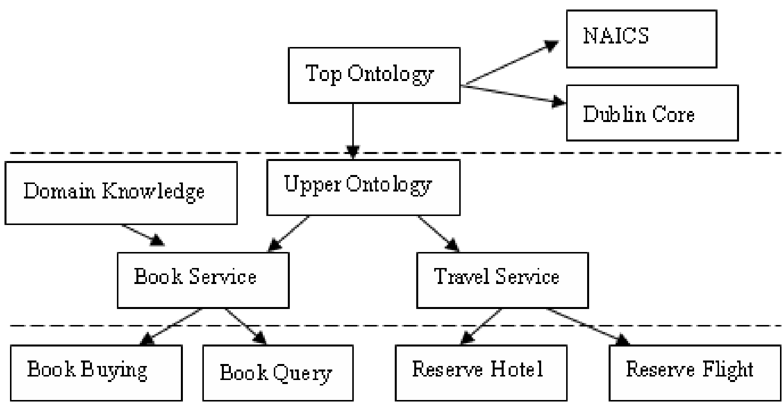

Fig. 1. A Layered Construction of Ontology

2.4 Quality Model

Definition 3 (Quality Model of Web Services). $QoS = \{t,c,a,s\ldots.\}$
Where:

- t: the response time of the certain Web service.
- c: the cost of the invoking of the certain Web service.
- a: the probability of availability of the certain Web service in a given interval, it is a statistic.
- s: the probability of success of the certain Web service in a given interval, it is also a statistic.

Vectors *t*, *a* and *s* are mature metric for quality of software, so their values are computed in the same way as that in quality of software [7].

(i) Vector *t* is numeric parameter, whose value is assigned when the ontology for a concrete service is built. For example, a provider assigns vector *t* according to the performance of his service when publishes his service, and a requester also assigns vector *t* according to his requirements when he summits his request.

(ii) Vector *a*, *s* can be computed by a counter in an *Agent* which is developed by the third party. Vector *a* is computed by the following formula, and Vector *s* can be computed by a similar way:

$$a = \frac{num(available\_invocation)}{num(total\_invocation)} \tag{1}$$

(iii) Vector *c* plays a critical role in discovery of SOC. Just like in the real world, the requesters pay more attentions on it than other vectors.

Definition 3 shows a general quality model. Considering the various domains, the model is defined as an open structure, where the users can define their own vectors about quality of service, and values of vectors vary with the domains. *tModel* in UDDI is exploited to support the model [8].

2.5 Implementation of Ontology

In order to implement the defined ontology, we select Protégé as the editor. Protégé is based on Java and provides a PnP environment. Protégé ontologies can be exported into a variety of formats including RDF (S), OWL, and XML Schema. Furthermore, RacerPro is a selection of inference engines as the back-end inference system (reasoner) for use with Protégé. The RacerPro system implements the description logic and can reason about OWL knowledge bases, together with some algebraic reasoning beyond the scope of OWL. With the help of Protégé, the defined framework of Service Ontology can be designed and the previous version of knowledge based can be exploited together.

3 An Example—Learning Resource Ontology

In this section, we will consider an example about service ontology. Learning Resources (for short: LR) have played an important role in modern education. While in terms of Web Services, all the items in the Internet, such software, hardware, information and so on, can be regarded as available *Services.* So is learning resources.

With reference to the main standard for learning object, we can define an ontology for LR compliance with *Definition* 2. *Table* 1 shows attributes of LR and related standards. Ontology LR is organized as shown in *Table* 1.

Table 1. Attributes of LR and Related Standard

<table>
<tr><th>Attribute</th><th>Sub-Element</th><th>Related Standard</th></tr>
<tr><td rowspan="7">d</td><td>identifier</td><td rowspan="14">IEEE LOM</td></tr>
<tr><td>title</td></tr>
<tr><td>language</td></tr>
<tr><td>keywords</td></tr>
<tr><td>description</td></tr>
<tr><td>version</td></tr>
<tr><td>contribute</td></tr>
<tr><td rowspan="6">f</td><td>format</td></tr>
<tr><td>size</td></tr>
<tr><td>requirement</td></tr>
<tr><td>duration</td></tr>
<tr><td>intended end</td></tr>
<tr><td>user role</td></tr>
<tr><td rowspan="6">p</td><td>cost</td></tr>
<tr><td>scientificity</td><td rowspan="3">CELTS-24</td></tr>
<tr><td>accessibility</td></tr>
<tr><td>update-interval</td></tr>
<tr><td>response time</td><td></td></tr>
<tr><td>probability of success</td><td></td></tr>
</table>

The following give more details about LR:

- LR lies in the concrete level of *Fig*. 1, just like *book buying*.
- IEEE LOM (Learning Object Metadata) [9] is a standard developed by IEEE, which lies in the domain level of *Fig*. 1.
- CELTS (Chinese E-learning Technology Standardization) [10] is a standard developed by Chinese government, and CELTS-24 is its sub-standard for Service Quality of e-Learning.
- Except attribute *response time* and *succeed-probability*, other attributes are assigned values compliance with the standards.

4 Discovery Process

The intention of Service Ontology defined in Section 2 is to add semantics to WSDL and UDDI. In this section, we show how to use Service Ontology to find the suitable service for requesters or compound service developers.

4.1 Discovery Process

(i) Transform the requests of customers in forms of Service Template compliance with Service Ontology. All the vectors *f*, *d*, *p* in *Definition* 2 as well as their sub-elements, will be used in the following match. It is necessary for compound service developers to decompose the request into atomic process.

(ii) Search engine in UDDI matches the requests against Service Advertisement. IOPE (Input, Output, Precondition, Effect) is the major factors in this stage. As a result, a candidate set of services is gained.

(iii) Analyze the performance constraints and order the selected services in the candidate set so that customers can select the most suitable services according to their own preference. Matching algorithm will be exploited, which is introduced in following section.

4.2 Similarity Matching

During the discovery process, several Agents are needed. Those Agents aim to implement matching algorithms. With the increase of Web Services specification in UDDI, it is difficult to find out a suitable service. The traditional retrieval technologies, such as keyword-based match, are not applicable. So is the classical set theory, whose logic is based on either 0 or 1. In this paper, *Fuzzy Theory* and *Similarity Function* [11,12] are selected to match *Service Template* against *Service Advertisement*. The result of match computed by *Formula* 2 is on closed interval [*0,1*]. This is a quantified analysis, whose result is more accurate than that of some other algorithms.

$$s(a,b) = \Sigma\ (\mu_i\ s_i(a,b)) \quad (\mu_i > 0, \sum \mu_i = 1) \tag{2}$$

Where:

- $s\ (a,\ b)$ is similarity between a and b, where a is the description given by a provider, b is that of a requester.
- μ_i is the weigh coefficient, they reflect the weightiness of attributes of service.

Apparently, it is easy to compute the value of similarity when IOPE is matched in step (ii) during discovery process. As for the match in step (iii), semantic relationships among services compliance with *Definition* 2 can be exploited so as to support the customers in discovery services that fit their requirements. A simplified form of Tversky function is applicable. We state more details about it in [13]. Furthermore, μ_i will be assigned by domain experts.

4.3 Implementation

The models and methods introduced are also used within SOA. This section states how to enable service discovery as described above.

4.3.1 Enhancing UDDI

Undoubtedly, all of the concrete services (mentioned in Section 2.3) are stored in UDDI Registry, while the existing UDDI Registry supports neither the description in OWL nor the performance vector in *Definition* 2. Therefore, one of our works is to extend the UDDI Registry, maintaining a full compatibility with it, so that the developers or the customers can either exploit the existing UDDI APIs or invoke the APIs provided.

Fortunately, one goal of UDDI is to make it easy to describe Web Services meaningfully, and its *tModel* structure provides the ability to describe compliance with certain specifications, concepts or other shared design. Besides *tModel*, UDDI v2 defines several elements for category: (i) *keyedReferenceGroup*, which is a simple list of *keyedReference* structures that logically belong together. It must contain a *tModelKey* attribute that specifies the structure and meaning of the *keyedReferences* contained in the *keyedReferenceGroup*. (ii) *catagoryBag*, which contains a list of business categories that each describes a specific business aspect of the *businessEntity*, such as industry category or a simple list of *keyedreferneceGroup*. It also can be used to categorize *tModel*. [14]

Making use of the extensive mechanism, UDDI can contain service description compliance with *Definition* 2. In the proposed method, we define a set of *tModel* to represent vector p in Definition 2, and some *tModel* for IOPE. All the sets are related with Service Ontology, while each *keyedReferernce* has its own *keyValue* that represents different ontology.

4.3.2 Annotated WSDL

Although Service Ontology is edited in OWL rather than OWL-S, WSDL is a de-facto standard for service functionality description. In the proposed method, we select Semantic Annotations for WSDL (for short: SAWSDL).

SAWSDL defines a set of extension attributes for WSDL that allows description of additional semantic of WSDL components. SAWSDL doesn't specify a language for representing the semantic models. Instead it provides mechanisms by which concepts from the semantic model can be referenced from within WSDL and XML Schema components using annotations. It also can be mapped into an RDF form compatible with previous versions. [15]

5 Conclusion and Further Work

An ontology for Semantic Web Services with both functional and non-functional attributes has been built in Protégé. By making use of the ontology in the model proposed in [13], much improvement will be made in Service discovery.

Service discovery is an ongoing research direction in Web Services community. Service Ontology is introduced to enrich the semantics of Web Service, including performance description. The proposed construction of Service Ontology makes it easy to identify and maintain the relationships among services. Exploiting such an ontology, as well as extension of UDDI and SAWSDL, service discovery techniques are used to match Service Template against Service Advertisement efficiently.

The further work includes:

- Do more competitive trials to prove the effect of the proposed method by recalling the services described in different way, such as WSDL, OWL-S and the proposed method.
- Do more study on mechanism in UDDI in order to enhance its semantics.
- Improve the match algorithm with reference to optimization technology in nonlinear systems.
- Optimize this method by take full advantage of traditional retrieval methods, such as Levenshtein Distance for string matching.

References

1. Fillman, F.R.: Semantic services. IEEE Internet Compute. 7 (2003)
2. Berners-Lee, T.: Semantic Web–XML2000 (2000), http://www.w3.org/2000/talks/ 1206-xml2k -th1/ silde10-0. html
3. Martin, D., et al.: OWL-S: Semantic Markup for Web Services (2004), http://www.w3.org/Submission/OWL-S/
4. Christensen, E., Curbera, F.: Web Services Description Language (WSDL)1.1. (2001), http://www.w3.org/TR/wsdl
5. Guarino, N.: Formal Ontology and Information Systems. In: Proceedings of FOIS 1998 (1998)
6. Ankolenkar, A., et al.: OWL-S: Semantic Markup for Web Services, OWL-S 1.0 Draft Release (2003)
7. Pressman, R.S.: Software Engineering: a Practitioner's Approach, 5th edn. China Machine Press (2005)
8. van Moorsel, A.: Metrics for the Internet Age: Quality of Experience and Quality of Business. In: Proceedings of the 5th Performability Workshop (2001)
9. Anonymity, Standard for Learning Object Metadata (2006), http://www.imsglobal.org/ metadata/mdv1p3/ imsmd_bestv1p3.html
10. Chinese E-Learning Technology Standardization Committee, Specification for Service Quality Management System of e-Learning (2003), http://www.celtsc. edu.cn/680751c665875e93
11. Guoen, Y., Rong, A.: A Review of Theories and Models of Similarity in Categorization. Psychological Exploration (2005)
12. Wang, Y., et al.: Semantic structure matching for assessing web-Service Similarity. In: Orlowska, M.E., Weerawarana, S., Papazoglou, M.M.P., Yang, J. (eds.) ICSOC 2003. LNCS, vol. 2910, Springer, Heidelberg (2003)
13. Qizhi, Q., et al.: Study on Ontology-based Web Services Discovery. In: Proceeding of the 11th International Conference on Computer Supported Cooperative Work in Design (2007)
14. Bellwood, T., et al.: UDDI: Version 3.0.2 (October 2004), http://uddi.org/pubs/uddi_v3.htm# _Toc85908023
15. Farrel, J., et al.: Semantic Annotations for WSDL and XML Schema (2007), http://www.w3.org/TR/sawsdl